.NET Overall Mock Questions

C# (please provide code examples)

1. What are the data types in C#? Is string a value type?
2. What is the Enumeration type? How to convert an enum type to a string? How to convert a string to an enum type? How to convert an enum to int?
3. What is the difference between **string** vs. **StringBuilder**?
4. What is the difference between **ref** & **out** parameters?
5. What is the difference between **constants** and **readonly**?
6. Difference between finally vs. Finalized()
7. Difference between Finalized() vs. Dispose()
8. Method overriding vs. method overloading? Can a private virtual method be overridden?
9. Interface vs. abstract class?
10. What are generics in c#?
11. What are delegates? How to create a delegate?
12. How to pass delegate as a method parameter?
13. Boxing vs. unboxing
14. What is OOP?
15. Compare var vs. object vs. dynamic.
16. How to check the type for dynamic?
17. What is nullable? How to assign a nullable to a non-nullable variable?
18. What is safe navigation in C#?
19. C# staticity:
    a. What is the **static** keyword? Can we use **this** keyword within a static method?
    b. What is a static constructor in C#?
    c. What does base() refer to? Can we use base() on static methods?
    d. What is a static class? Can we derive a static class?
    e. What is the difference between static class and sealed class?
20. Difference between Hashtable and Dictionary.
21. How to create a Generic class and a Generic interface?
22. What is the **yield** operator?
23. What is the **deconstructor**? Will you use a deconstructor?
24. Lazy binding vs. late binding.
25. What are anonymous in C#?

SQL Server(please provide examples where it makes sense)

1. What is an index?
    a. Difference between clustered index and non-clustered index
2. In SQL Server, how to store and manipulate data in a temporary manner? Please provide code examples. (**Table Variables, Temporary Tables, CTE, and their differences**)
3. What is a view? Please give an example of how to create a **view** in SQL Server.

4. Explain Stored Procedures.
5. Delete vs. Truncate.
6. What is a composite key?
7. What is data integrity?
8. Explain ACID.
9. Have you used cursor?
10. Anonymization.
11. Normalization and its advantages.
12. How complex have you used SQL? (can talk about sub-queries, joins, view, SP…)
13. Given an Employee table with ID, Department, and Salary, write a SQL query for the following:
    a. Find the number of employees in each department.
    b. Get the highest salary per department group.
    c. Find the employees who have the top salary in each department.
    d. Find all employees with the 3rd highest salary.

LINQ (please provide code examples)

1. Difference between IEnumerable and IQueryable
2. What are extension methods?
3. Single() vs. First() vs. Take(1)
4. When to use First() and when to use FirstOrDefault()?
5. How to join on multiple conditions?
6. What is Distinct()?
7. Contains() vs. Take() vs. TakeWhile()
8. Select() vs. SelectMany()
9. Assume you have a List<Employee> and a List<Department>
    a. Sort the employees by their last names in ascending order
    b. Join two lists and return the result as a new list of a custom type {Employee, Department}
    c. How do you perform a left outer join between the employee and department list? (GroupJoin(), SelectMany())

ADO.NET

1. What is ADO.NET?
2. What are the key components of ADO.NET?
3. How do you use ADO.NET? (the steps)
4. What are the uses of **using** statement?
5. ExecuteNonQuery() vs. ExecuteScalar(), ExecuteReader()
6. What is DataAdapter? How does it work? What are DataSet and DataTable?
7. Fill() vs. Update()
8. How to use store procedure in ADO.NET?
9. How to use transactions in ADO.NET?
10. What are the two types of transactions supported by ADO.NET?

11. What’s the difference between DataReader and DataSet?
12. What is SQL Injection?
13. How to prevent SQL Injection when using ADO.NET?

ASP.NET Core

1. ASP.NET Core vs. ASP.NET.
2. What is ASP.NET Core?
3. Can you explain the role of Startup.cs? (now we only have a Program.cs file, but you need to know the previous version)
4. What’s the use of the ConfigureServices() method in Startup.cs.
5. What’s the use of the Configure() method in Startup.cs.
6. Can you explain the middleware pipeline?
7. Explain how routing works.
    a. Conventional routing
    b. Attribute routing
8. Have you used the MVC design pattern?
9. How do you implement Authentication and Authorization in ASP.NET Core MVC?
10. What does the [Authorize] attribute do?
11. What is ModelState in MVC?
12. What is the MVC lifecycle? Request lifecycle.
13. Razor Pages vs. MVC.
14. What’s the difference between ActionResult and IActionResult?
15. What is IoC (Inversion of Control)?
16. How does dependency injection work in ASP.NET Core, and how can you use it in your applications?
17. How many approaches do we have to pass data from the controller to a view?
18. How to pass data from the view to the controller?
19. Explain REST APIs.
20. How to build REST APIs?
21. How to consume REST APIs?
22. POST vs. PUT vs. PATCH. GET vs. POST.
23. Meaning of HTTP Status Code – 200, 302, 400, 401, 402, 500, 503
24. How to pass data from client to server?
25. Are you exposed to APIs or services from 3rd party? How did you do that?
26. How to do Validation for your Web API.
27. REST API vs. GraphQL.
28. What is CORS? How to deal with CORS?
29. How do you document your endpoints?

EF Core

1. What is ORM, and what are its benefits?
2. Have you used Entity Framework Core?
3. What are the loading strategies for fetching data in EF Core?

    a. What’s the default loading strategy?
    b. What’s the keyword for enabling eager loading?
4. What is the difference between deferred execution and lazy loading?
5. How to use store procedures in EF Core?
6. How to use transactions in EF Core?
7. How to prevent SQL Injection when using EF Core?
8. How to handle errors in Stored Procedures?
9. What is a navigation property?
10. What is fluent API convention?
11. How to do Many-to-Many mapping in EF Core?
12. What is SaveChanges() in EF Core?
13. How do you track DB changes in EF Core?

Angular

1. What is DOM?
2. What are the components, services, and directives?
3. What is Pipe? Custom pipe/directive
4. How to transfer data between two angular components?
5. How to transfer data from template to component?
6. How is dependency injection applied?
7. How to fire a call to the server?

Azure and Cloud Service

1. Any Experience with Azure?
    a. Azure SQL Database
    b. Azure Blob Storage
    c. Azure Service Bus
    d. Azure DevOps
    e. Azure Key Vault…

Microservices

1. Difference between Monolithic vs. Microservices
    a. Advantages and disadvantages of Monoliths and Microservices
2. What is cascading failure? How to prevent this failure?
3. What is fault tolerance? How to make your microservice fault tolerant?
4. How do microservices communicate?
5. What is the Gateway? Is it necessary?
6. Do you have any experience with messaging? (Rabbit MQ and Azure Service Bus)
    a. What are the components of RabbitMQ?
    b. What are the different types of exchanges that exist in Rabbit MQ

Other Topics

1. Difference between .NET Framework and .NET Core.
2. Have you worked with any design principles? (SOLID)
3. Have you used any design patterns?
   a. Singleton Design Pattern
      i. How would you implement a singleton in C#?
   b. Factory Design Pattern
   c. Abstract Factory Design Pattern
   d. MVC Design Pattern
   e. Repository Design Pattern
   f. Unit of Work Design Pattern (vs. Repository Design Pattern)
4. How do you manage authentication and authorization in your application? How do you secure your APIs? (talk about the JWT flow)
5. How to handle exceptions in your web application?
6. Have you worked with the agile methodology?
7. How did you deploy your application?
8. What is CI/CD? How did you use CI/CD in your application?
9. Have you used Git?（Git workflow）
   a. List the commands you used and what's the meaning behind them.
10. What is type-safe?
11. What is the correct workflow of the following processes in CI/CD?
    a. Check of Code
    b. Build
    c. Running Unit Testing
    d. Running Static analysis
12. What are managed code and unmanaged code？
13. What is Pair Programming
14. What tool do you use to test the functionality of your code?
15. Please give some use cases for Docker.
16. Can you talk about how internet users can grant websites access to certain information or permissions?
17. How to debug your application?
18. The production support team reports that your application is running slow as the customer base becomes large, what can you do to increase the performance of your application? (async, caching…)